5П

УДК 519.9

В.Н. Петрович
Киевский национальный университет имени Тараса Шевченко, г. Киев, Украина
filonval@i.com.ua

Идентификация параметров математических моделей динамических систем управления

В статье рассматривается программный комплекс для оценивания параметров математических моделей сложных динамических систем на основе экспериментальных данных, полученных в процессе их функционирования. Математической базой данной системы являются современные методы оптимизации, рекуррентные алгоритмы оценивания, методы линейной алгебры, методы линейного и нелинейного регрессионного анализа, методы вычислительной математики. Комплекс использует объектно-ориентированный подход при реализации программ, имеет удобный интерфейс, не требует от пользователя знания особенностей языка программирования или деталей схемы организации процесса вычислений в комплексе.

Введение

На современном этапе развития производства и конструирования сложных динамических объектов невозможно обойтись без предварительной разработки математической модели такого объекта. Для тестовых испытаний строятся сложные математические модели (что, как правило, задаются системами дифференциальных уравнений). Такие модели на практике очень сложные и содержат много параметров и параметр управления, с помощью изменения которого возможно тестирование модели при различных условиях. Ясно, что важнейшая задача такого моделирования состоит в том, чтобы как можно точнее описать модель. Но не всегда при описании моделей можно определить значение некоторых параметров. Примером таких величин могут быть внешние силы природы, действующие на поверхность самолета, внутренние волновые потоки. То есть те величины, которые не может контролировать человек. Одной из реальных задач идентификации была идентификация параметров модели продольного движения самолета АН [1]. Практически можно было исследовать влияние рулевого крыла на динамику полета самолета и моделировать движение и при этом находить такие параметры, как аэродинамический момент, подъемную силу, сопротивление внешней среды.

Следовательно, задача заключается в том, чтобы по экспериментальным данным приблизить математическую модель к реальной с помощью критерия. Как показывает практика, методы минимизации такого критерия зависят от структуры самого критерия, от общего вида системы. Итак, для моделирования и исследований необходимо иметь достаточно широкий спектр методов минимизации, имеющих различные характеристики скорости сходимости. Каждый из существующих алгоритмов многомерной оптимизации имеет свои преимущества и недостатки. Поэтому при решении задачи минимизации функции был использован разработанный комплекс алгоритмов минимизации, последовательно подключающий различные алгоритмы на тех этапах процесса оптимизации, где каждый из них является наиболее эффективным. Существенной помощью в выборе эффективного метода было графическое изображение функции-критерия.

Как видно, задача идентификации приведена к задаче минимизации, что является само по себе достаточно сложной задачей. Но на данном этапе возникает проблема –

достаточно неудобно было бы для каждой новой системы или нового метода минимизации перекомпилировать проект. **Цель работы** заключается в том, чтобы создать такой комплекс алгоритмов, который бы легко адаптировался под разные системы и разные методы. Каждый пользователь может написать свой метод или описать новую систему, оформить в виде DLL и тогда автоматически этот метод или система будет заметна программе. Естественно, все методы и системы должны подчиняться соответствующему шаблону. Таким шаблоном выступают базовые классы, от которых пользователь должен наследовать свой класс. Для примеров и тестирования некоторых систем реализованы методы одномерной, многомерной минимизации.

Поскольку на практике мы имеем достаточно сложные системы, невозможно заранее прогнозировать, насколько корректным будет решение и какие методы необходимо использовать. Как видно, решением будут минимизированы параметры для функции критерия, поэтому понятно, что для того чтобы подобрать удачные методы минимизации нужно знать топологическую структуру функции критерия (ее графическое представление). Поэтому была реализована возможность посмотреть на функцию в трехмерном пространстве, фиксируя другие переменные. Такая предварительная информация позволяет увидеть, какая форма критерия – выпуклая, овражная или может существовать несколько экстремальных точек – по этой информации можно подобрать метод, как лучше решать данную задачу.

С помощью данной системы был решен ряд тестовых примеров, которые показали ее работоспособность и удобство подключения новых систем. Также была взята реальная система, описывающая продольное движение летательного аппарата, и подсчитаны параметры на реальных данных.

Постановка задачи и критерии оценивания

Рассмотрим постановку задачи идентификации для динамических систем, описываемых обыкновенными дифференциальными уравнениями, правые части которых нелинейные относительно параметров и фазовых координат, а также проведем классификацию задач и алгоритмов их решения в зависимости от классов модели и критериев идентификации. Запишем общий вид модели

$$\frac{dx}{dt} = F(x(t), p, t). \tag{1}$$

Начальные условия для задачи Коши $x^m(t_0) = x_0^m$. Вектор параметров $p = (p_1, \ldots, p_m)$. Вектор модельных фазовых координат

$$x^m(t) = \left(x_1^m(t), \ldots, x_n^m(t)\right).$$

Вектор экспериментальных фазовых координат

$$x^e(t) = \left(x_1^e(t), \ldots, x_n^e(t)\right).$$

На основе экспериментальных данных (наблюдений) за объектом на интервале времени $[t_0 + (k-1)T,\ t_0 + kT]$, $k = 1, \ldots, r$ на каждом из таких промежутков фиксируются экспериментальные значения фазовых переменных, обозначим их как x^e, где $t_j = t_{j-1} + \Delta t \quad j = 1, \ldots, N$; $\Delta t = T / N \quad j = 1, \ldots, N$.

Необходимо идентифицировать (определить) параметры p из условия соответствия выбранной модели (1) данному объекту по некоторому критерию качества модели

$I(p)$ так, чтобы $I(p^*) = \min_p I(p)$, при этом параметры p могут быть произвольными или иметь какие-то ограничения $p \in D = \{p : a_i \le p_i \le b_i\}$.

В зависимости от требований, предъявляемых к задаче, выбирается соответствующий критерий качества $I(p)$, который может задаваться разными способами:

1. В виде функции среднеквадратичного критерия:

$I(p) = \int_{t_0}^{t_0+T} \left\| x^m(t,p) - x^e(t) \right\| dt$, $\|\ \|$ – есть расхождение модельными фазовыми координатами и фазовыми координатами экспериментальных данных, $\|\ \|$ – норма в Евклидовом пространстве, соответствующая ему дискретная форма

$$I(p) = \sum_{j=1}^{N} \sum_{i=1}^{n} [x^m(t_j, p) - x^e(t_j)]^2 .$$

2. В виде минимаксной функции несогласованности экспериментальных и модельных фазовых координат:

1) $I(p) = \max_{1 <= j <= N} \| x^m(t_j, p) - x^e(t_j) \|$ – сумма по всем максимальным отклонениям;

2) $I(p) = \sum_{i=1}^{n} [\max_{1 <= j <= N} |x_i^m(t_j, p) - x_i^e(t_j)|]$ – максимум по всем максимальных отклонениям. Как видно, здесь использовано Чебышевскую норму.

3. В виде функции-невязки, которая в случае линейной функции правых частей системы (1) относительно параметров p позволяет свести задачу к линейной идентификации (в данной работе линейный случай рассматриваться не будет)

$$I(p) = \left\| \frac{dx^e}{dt} - F(x^e, t, p) \right\|^2, \quad I(p) = \sum_{j=1}^{N} \sum_{i=1}^{n} [x_i^e{}'(t_j) - F_i(x_i^e(t_j), t_j, p)]^2 .$$

Выбор конкретного критерия оценки $I(p)$ зависит от характера данных, от класса модели и требований, предъявляемых к алгоритму.

При выборе первого критерия фактически решается задача среднеквадратичной аппроксимации данных эксперимента $x^e(t)$ решениями задачи Коши (1) и (2), то есть значениями $x^m(t,p)$. Для произвольных типов моделей (1) этот критерий является наиболее удобным с точки зрения различия $x^m(t,p)$ и $x^e(t)$. Функция $I(p)$ этого критерия является выпуклой, что гарантирует хорошую сходимость некоторых методов минимизации. Эта функция дифференцирована, позволяет применять градиентные методы спуска, которые являются достаточно удобными в вычислении. Но при применении методов поиска минимума гарантировано большое количество вычислений функции $I(p)$, а в некоторых методах первой или второй производной, что делает часть этих методов совсем несостоятельными к решению таких задач, поскольку вычисления этой функции для каждой компоненты вектора параметров p требует решения задач задачи Коши на интервале $[t_0,\ t_0 + T]$.

При выборе второго критерия фактически строится Чебышевское приближение (равномерное приближение) экспериментальных данных $x^e(t)$ и результатов задачи Коши $x^m(t,p)$, минимизирует максимальное отклонение модельных решений и экспериментальных наблюдений на интервале $[t_0,\ t_0 + T]$. В некоторых случаях

практическое использование этого критерия может быть не эффективным. Когда наблюдения $x^e(t)$ наблюдаются на фоне случайных возмущений с большими выбросами, в этом случае необходимо предварительное сглаживание экспериментальных данных. Методы решения задачи идентификации при этом критерии, а также трудности использования алгоритмов остаются те же.

Второй проблемой этого критерия является проблема недифференцируемости функции $I(p)$, что заставляет использовать различные методы к нахождению градиента функции.

Если рассмотреть различные виды дискретной формы, то можно сказать, что критерий типа $I(p) = \sum_{i=1}^{n} [\max_{1<=j<=N} |x_i^m(t_j, p) - x_i^e(t_j)|]$ является достаточно целесообразным, поскольку минимизируется сумма по всем максимальным отклонениям, а значит, гарантирует уменьшение максимальных отклонений для всех максимальных отклонений.

Для функции $I(p) = \max_{1 \le i \le n} [\max_{1 \le j \le N} | x_i^m(t_j, p) - x_i^e(t_j) |]$ берем максимум по всем максимальным отклонениям, а значит, уменьшать будем только само максимальное отклонение.

Третий критерий наиболее распространен в инженерной практике. При этом критерии фактически решается задача среднеквадратичной аппроксимации производных экспериментальных данных функциями правых частей. В случае линейной по параметрам модели этот критерий позволяет свести задачу нахождения параметров p к линейной задаче, то есть к системе линейных уравнений $Ap = b$, менее сложной задаче. Для этого критерия также есть трудности, во-первых, задача дифференцирования экспериментальных данных, которые наблюдаются, – это некорректная задача, во-вторых, во многих случаях система $Ap = b$ имеет плохо обусловленную матрицу, а значит, решения могут быть как угодно далекими от истинных. Использование специальных методов решения линейных систем, имеющих плохо обусловленную матрицу, требуют расчетов с более высокой точностью, тем самым подавляя точность экспериментальных данных.

Общая характеристика алгоритмов нелинейной идентификации

Общий алгоритм решения поставленной задачи предусматривает такие этапы:

1. Найти решения системы дифференциальных уравнений (1) для начального (априорного) значения параметра $p = (p_1, ..., p_n)$, для начальных условий задачи Коши (2) на промежутке $[t_0, t_0 + T]$.

2. Для данных значений $x^m(t, p)$ – решений задачи Коши, $x^e(t)$ – экспериментальных данных найти значение функции – критерия качества $I(p)$, которая была описана выше.

3. Найти $\bar{p} = \arg\min_{p \in E^m} I(p)$, найденное p – будет следующим значением параметра для решения задачи Коши.

Поскольку задача состоит в нахождении неизвестных параметров нелинейной системы дифференциальных уравнений (1) как относительно самого параметра, так

и относительно фазовых переменных, то основной задачей является задача многомерной минимизации. Сформулируем задачу безусловной минимизации функции многих переменных, которая выбрана из одного из критериев качества, приведенных выше.

Для решения задачи минимизации нахождение точки минимума функции применяются методы, которые включают в себя методы дифференцирования (а вернее, нахождение градиента). Если описать вычислительную схему – большинство алгоритмов можно выделить в ряд последовательных этапов, а именно:

1) выбор начального приближения параметров p_0 и вычисления $I(p_0)$;

2) определение направления спуска p_k для минимизации $I(p)$ (способ решения этой подзадачи является определяющим для каждого конкретного алгоритма) характерна своя стратегия при выборе направления спуска p_k фактически – это нахождение градиента функции $I(p)$;

3) определение точки минимума $\lambda_{\min}$ на луче $p = p_k + \lambda * h_k * g_k$, где $\min I(p_k + \lambda * h_k * g_k) = I(p_k + \lambda_{\min} * h_k * g_k)$;

4) переопределение полученной точки минимума на луче $p_{k+1=} p_k + \lambda * h_k * g_k$ в качестве исходной на следующем этапе: $p_{k+1} = p_k + \lambda * h_k * g_k$.

Условный переход на 2), если критерий выхода из алгоритма не выполняется. Однократное выполнение этапов 2 – 4 будем называть одной итерацией процесса минимизации. На каждом из этапов нужно фактически решить некоторую самостоятельную подзадачу. Для решения подзадач 1 и 3 можно использовать несколько эквивалентных (в смысле решаемой нами задачи) методов, которые реализованы в виде самостоятельных программных модулей.

Для отдельной задачи минимизации выбор начальной точки x_0 осуществляется управляющей программой минимизации. Если начальная точка не задана, то вместо нее в качестве исходной точки для поиска выбирается случайная точка, что равномерно распределена в некоторой заданной области D_0 в n-мерном параллелепипеде $D = (x : a < x < b)$.

Для случая критерия качества равномерного приближения мы имеем недифференцированную функцию $I(p) = \max_{1<=j<=N} \max_{1<=i<=n} | x_i^m(t_j, p) - x_i^e(t_j) |$. Невозможно использование прямых методов нахождения $\bar{p} = \arg\min_{p \in E^m} I(p)$ с помощью градиентов, поскольку функция имеет вид $F(tj) = \max_{1<=i<=n} | x_i^m(t_j, p) - x_i^e(t_j) |$ (рис. 1).

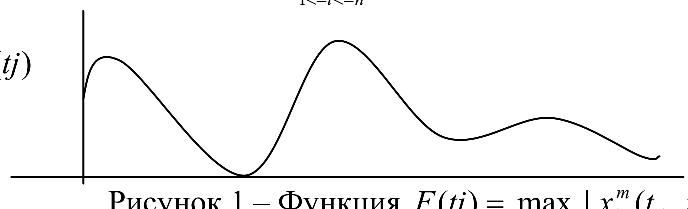


Рисунок 1 – Функция $F(tj) = \max_{1<=i<=n} | x_i^m(t_j, p) - x_i^e(t_j) |$

Здесь много экстремумов, следовательно, нахождение градиента (направления спуска) для одного максимума не гарантирует спуск другого максимума. Поэтому решение проблемы заключается в том, чтобы выделить максимумы $F\max 1(t_1), \ldots, F\max n(t_n)$ для каждого такого промежутка (обозначим, k-й промежуток), решить задачу Коши (1) на промежутке $[t_0, t_0 + t_k]$. Найти для этих решений $I(p)$, найти для этого промежутка свое направление спуска p_k.

Нахождение направлений спуска (на каждом из таких интервалов) уже позволит применять градиентные методы, которые мы описали выше. Пусть нашли такие векторы направлений спуска для каждого из S интервалов: $p_1,\dots,p_s$. Найдем теперь для линейной оболочки, построенной на векторах: $p_1,\dots,p_s$, такой вектор x, который будет принадлежать этой оболочке, и будет наименее удален от начала координат.

Пусть для каждого промежутка нашли вектор наискорейшего спуска p_i (для i-го промежутка) строим вектор x, который является линейной комбинацией векторов p_i $x = k_1 p_1 + \dots + k_s p_s$.

Решаем задачу минимизации

$$\| x \| = (x,x) = k1^2 \,\| p1 \| + k2 * k1(p1,p2) + \dots + ks^2 \,\| ps \| \to \min,$$

если $\sum_{i=1}^{s} k_i = 1$. Для этой задачи строим функцию Лагранжа

$$L(\lambda) = k1^2 \,\| p1 \| + k2 * k1(p1,p2) + \dots + ks^2 \,\| ps \|$$

$$+ \lambda * (\sum_{i=1}^{s} k_i - k1^2 \,\| p1 \| + k2 * k1(p1,p2) + \dots + ks^2 \,\| ps \|),$$

$$\frac{\partial L(\lambda, k1,..ks)}{\partial k_i} = 0; \quad \frac{\partial L(\lambda, k1,..ks)}{\partial \lambda} = 0;$$

По сути, это линейная система уравнений. Ее решения относительно λ и дадут значение искомого вектора x.

Итак, если есть векторы $p_1,\dots,p_n$, то для построения вектора $x = k_1 p_1 + \dots + k_n p_n$, при условии, что $\|x\| \to \min$ и $\sum_{i=1}^{s} k_i = 1$. Если решить систему, получим

$$\lambda = -\frac{2}{\left(\frac{1}{\|p_1\|} + \dots + \frac{1}{\|p_n\|}\right)} \qquad k_i = -\frac{\lambda}{2\|p_i\|}.$$

Рассмотрим подробнее подзадачу 4. В ней понятие «критерий» включает в себя два условия:

1) алгоритм «исчерпал себя», что отразилось на выполнении так называемого «внутреннего» критерия (этот критерий свой для каждого алгоритма), тогда выход из программы минимизации происходит с кодом возврата $kw = 1$.

2) было выполнено заданное число KIT итераций, $MinM$ – это экземпляр класса многомерной задачи минимизации, однако внутренний критерий не исполнился, тогда возврат происходит с $kw = 2$, число $MinM -> iKIT$ задается пользователем или по умолчанию выбирается управляющей программой. При возвращении в управляющую программу минимизации с кодом $kw = 2$ осуществляется анализ скорости сходимости поискового алгоритма с помощью подпрограммы $stban$.

Если сбор информации автоматизирован, то можно придерживаться таких оценок сходимости. Если не выполняются условия

$$R_x = \frac{1}{iMit} \frac{\left[\sum_{k=1}^{iMit}\left(\sum_{j=1}^{m}(x_k^i - x_{k+1}^i)^2\right)\right]^{1/2}}{VA}, \quad R_p = \frac{1}{iMit} \frac{\left[\sum_{k=1}^{iMit}\left(\sum_{j=1}^{m}(x_k^i - x_{k+1}^i)^2\right)\right]^{1/2}}{VK}$$

то скорость сходимости данного алгоритма неудовлетворительная и пользователь должен изменить методы минимизации. В случае неудовлетворительной сходимости пользователь также может управлять алгоритмами, изменяя значения их входных параметров или конфигурацию модулей, решающих задачу.

Заключение

Разработана усовершенствованная версия этой системы для персональных компьютеров в среде Windows на C++, в которой представлены более 15 методов оценивания параметров и состояний, позволяющих решать как линейные, так и нелинейные задачи идентификации в условиях «зашумленности» и неполного поля измерений, включая возможность обработки данных в режиме реального времени. Возможна переориентация комплекса на решение задач идентификации для других предметных областей применения. Данная система позволяет решать задачи идентификации для динамических систем, математическая модель которых описывается системами обыкновенных дифференциальных уравнений или функциональными соотношениями. В системе предусмотрен удобный интерфейс с пользователем на всех этапах решения задачи, удобная графическая поддержка процесса решения задач.

Литература

1. Сучасні методи і інформаційні технології математичного моделювання, аналізу й оптимізації складних систем / [Ф.Г. Гаращенко, М.Ф. Кириченко, Ю.В. Крак та ін.]. – К. : ВПЦ «Київський університет». – 2006. – 200 с.
2. Пашковский И.М. Летные испытания самолетов и обработка результатов испытаний / Пашковский И.М., Леонов В.А., Поплавский Б.К. – М. : Машиностроение, 1985. – 416 с.
3. Лоусон Ч. Численное решение задач метода наименьших квадратов / [Ч. Лоусон, Р. Хенсон ; пер. с англ.]. – М. : Наука, 1986. – 232 с.

Literatura

1. Suchasni metodi i informaciini tehnologiï matematichnogo modelyuvannya, analizu i optimizaciï skladnih sistem / [F.G. Garaschenko, M.F. Kirichenko, Yu.V. Krak ta in.]. – K. : VPC «Kiïvs'kii universitet». –2006. – 200 s.
2. Pashkovskii I.M. Letnye ispytaniya samoletov i obrabotka rezul'tatov ispytanii / Pashkovskii I.M., Leonov V.A., Poplavskii B.K. – M. : Mashinostroenie, 1985. – 416 s.
3. Louson Ch. Chislennoe reshenie zadach metoda naimen'shih kvadratov / [Ch. Louson, R. Henson ; per. s angl.]. – M. : Nauka, 1986. – 232 s.

Петрович В.М.
Ідентифікація параметрів математичних моделей динамічних систем керування
У статті розглядається програмний комплекс для оцінювання параметрів математичних моделей складних динамічних систем на основі експериментальних даних, отриманих в процесі їх функціонування. Математичною базою даної системи є сучасні методи оптимізації, рекурентні алгоритми оцінювання, методи лінійної алгебри, методи лінійного і нелінійного регресійного аналізу, методи обчислювальної математики. Комплекс використовує об'єктно-орієнтований підхід при реалізації програм, має зручний інтерфейс, не вимагає від користувача знання особливостей мови програмування або деталей схеми організації процесу обчислень в комплексі.

Petrovich V.M.
Identification of Parameters of Mathematical Models of Dynamic Control Systems
The article discusses a software package for estimating the parameters of mathematical models of complex dynamical systems based on experimental data obtained in the course of their operation. The mathematical basis of this system are the modern methods of optimization, recursive estimation algorithms, linear algebra, methods of linear and nonlinear regression analysis, methods of computational mathematics. Complex uses object-oriented approach in the implementation of programs, has a convenient interface that does not require any knowledge of programming language or the details of the schemes of calculations in the complex.

Статья поступила в редакцию 06.06.2011.